

## Taller 1 – Parte 1

### Preparación del ambiente de trabajo

#### Objetivo general

El objetivo de este taller es preparar nuestro ambiente de trabajo de cara al módulo 2. Es importante que realice los siguientes pasos en el computador en que vaya a desarrollar su trabajo. Si necesita trabajar en otro computador, es muy recomendado seguir de nuevo estos pasos para no tener problemas.

#### Punto 1 | PIP

PIP (Package Installer for Python) es el manejador de paquetes por defecto de Python. Un manejador de paquetes, es un software que permite instalar librerías y complementos para modificar nuestro entorno básico de Python. Es normal necesitar instalar paquetes cuando se desarrolla software, en particular cuando se utilizan tecnologías de vanguardia. Esta vez instalaremos un paquete básico que no supone ningún peligro para nuestro computador.

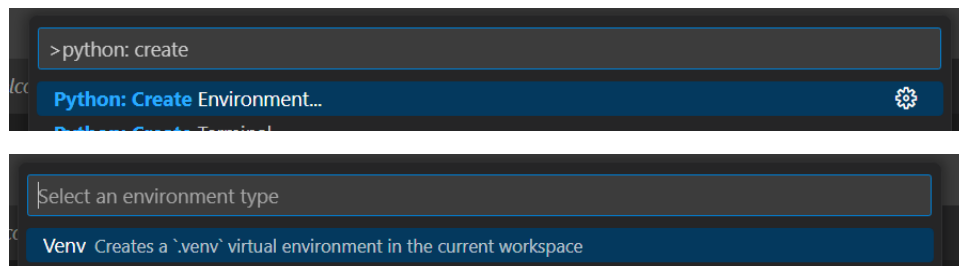
Geopy es una librería con funciones que permiten localizar geográficamente coordenadas, obtener el nombre de un lugar dadas sus coordenadas, entre otras funciones. Vamos a instalar Geopy para ver cómo funciona PIP.

```
pip install geopy
```

Con el comando **pip install** podremos instalar una librería en específico, en este caso es Geopy. Abra su terminal en VSCode y ejecute el comando para instalar la librería. Pasados unos minutos, ahora contará con geopy en su computador.

#### Punto 2 | Un entorno virtual

Hay situaciones en que la instalación de algún paquete o software puede generar conflictos de versiones con otro software. La instalación de frameworks y de otras librerías pesadas es un buen ejemplo de software que potencialmente puede causar conflictos con la configuración base del computador. La solución para estos casos es la creación de un entorno virtual. En primer lugar, es necesario abrir una carpeta en específico en visual studio code, en esta se guardarán todos los archivos solicitados. Para crear un entorno virtual seleccione View>Command Palette y seleccione las siguientes opciones:



Con esto habrá creado un entorno virtual en el cual podrá instalar paquetes y librerías sin ninguna preocupación. También es útil si se tienen varias versiones de un mismo software que pueden entrar en alguna clase de conflicto.

### Punto 3 | Flask

Ya que hemos creado nuestro entorno virtual. El siguiente paso es descargar la librería que vamos a utilizar en nuestro curso. Como lo vimos en la clase de hoy, Flask es uno de los frameworks existentes para trabajar en backend en python. Estando dentro de nuestro entorno virtual, procederemos a instalar Flask con el comando:

```
pip install flask
```

Esto instalará la librería con el framework de Flask y todas las demás librerías dependientes que nos pueden ser de ayuda.

### Punto 4 | Hola Mundo

El último paso es probar que todo funcione. Copie el siguiente programa en un archivo de python llamado **app.py**. Guárdelo en la carpeta del entorno virtual.

```
app.py
env > app.py > ...
1  from flask import Flask
2  app = Flask(__name__)
3
4  @app.route("/")
5  def home():
6      return "Hello, Flask!"
```

Ahora desde la terminal use el comando **python -m flask run**. Esto generará un letrero de parecido a este.

```
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Abra la dirección <http://127.0.0.1:5000> en un navegador de su preferencia. Observe el resultado.

### Entrega

No es necesario realizar una entrega. Guarde el trabajo desarrollado el día de hoy pues será fundamental y necesario para la parte 2 de este taller.